Online Quiz Portal Using spring boot and RestApi

MAIN SPRING APPLICATION CLASS

```
package com;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;


@SpringBootApplication
@ComponentScan({"com.entity","com.repo","com.controller","com.service"})
public class QuizPortalRestApiApplication {

	public static void main(String[] args) {
		SpringApplication.run(QuizPortalRestApiApplication.class, args);
	}

}
```

ENTITY CLASSES

Admin

```
package com.entity;

import org.springframework.stereotype.Component;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Component
@Entity
@Table(name="admin")
public class Admin {

	@Id
	private int id;
	@Column(name="Uname")
	private String uname;

	@Column(name="Pwd")
	private String pwd;


	public Admin() {}
	@Override
	public String toString() { return "Admin [id=" + id + ", uname=" + uname + ", pwd=" + pwd +
"]";}

	public Admin(int id, String uname, String pwd) {super(); this.id = id;this.uname =
uname;this.pwd = pwd;}
	public int getId() {return id;}
	public void setId(int id) {this.id = id;}
	public String getUname() {return uname;}
	public void setUname(String uname) {this.uname = uname;}
	public String getPwd() {return pwd;}
	public void setPwd(String pwd) {this.pwd = pwd;}
}
```

User

```
package com.entity;

import java.io.Externalizable;
import java.io.IOException;
import java.io.ObjectInput;
import java.io.ObjectOutput;

import org.springframework.stereotype.Component;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
@Component
@Entity
@Table(name="Userss")
public class User implements Externalizable {
	@Id
	private Integer uid;

	@Column(name="Email")
	private String emailid;

	@Column(name="upwd")
	private String password;

	@Column(name="Ph_NO")
	private long phno;

		public User() {}

		public User(int uid, String emailid, String password, long phno) {super();this.uid =
uid;this.emailid = emailid;this.password = password;this.phno = phno;}

		public int getUid() {return uid;}

		public void setUid(int uid) {this.uid = uid;}

		public String getEmailid() {return emailid;}

		public void setEmailid(String emailid) { this.emailid = emailid;}

		public String getPassword() {return password;}

		public void setPassword(String password) { this.password = password;}

		public long getPhno() {return phno;}

		public void setPhno(long phno) { this.phno = phno;}

		@Override
		public void writeExternal(ObjectOutput out) throws IOException {}

		@Override
		public void readExternal(ObjectInput in) throws IOException, ClassNotFoundException
{}

}
```

Question

```
package com.entity;
import org.springframework.stereotype.Component;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Component
@Entity
@Table(name="question")
public class Question {

	@Id
	private int qid;
	@Column(name="question")
	private String quest;
	private String opt1;
	private String opt2;
	private String opt3;
	private String opt4;
	private int ans;

	public Question() {}


	public Question(int qid, String quest, String opt1, String opt2, String opt3, String opt4, int ans) {
		super();
		this.qid = qid;
		this.quest = quest;
		this.opt1 = opt1;
		this.opt2 = opt2;
		this.opt3 = opt3;
		this.opt4 = opt4;
		this.ans = ans;
	}
	public int getQid() {return qid;}
	public void setQid(int qid) {this.qid = qid;}
	public String getQuest() {return quest;}
	public void setQuest(String quest) {this.quest = quest;}
	public String getOpt1() {return opt1;}
	public void setOpt1(String opt1) {this.opt1 = opt1;}
	public String getOpt2() {return opt2;}
	public void setOpt2(String opt2) {this.opt2 = opt2;}
	public String getOpt3() {return opt3;}
	public void setOpt3(String opt3) {this.opt3 = opt3;}
	public String getOpt4() {return opt4;}
	public void setOpt4(String opt4) {this.opt4 = opt4;}
	public int getAns() {return ans;}
	public void setAns(int ans) {this.ans = ans;
	}
	@Override
	public String toString() {
		return "Question [qid=" + qid + ", quest=" + quest + ", opt1=" + opt1 + ", opt2=" + opt2 +
", opt3=" + opt3
				+ ", opt4=" + opt4 + ", ans=" + ans + "]";
	}


}
```

Quiz

```
package com.entity;
import org.springframework.stereotype.Component;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;

@Component
@Entity
@Table(name="quiz")
public class Quiz {

	@Id
	private int quid;

	private String title;
	private int quizno;
	@Column(name="Subject")
	private String subject;

	@ManyToOne
	@JoinColumn(referencedColumnName = "qid")
	private Question qid;
	public Quiz() {}


	public Quiz(int quid, String title, int quizno, String subject, Question qid) {super();this.quid = 
quid;this.title = title;this.quizno = quizno;this.subject = subject;this.qid = qid;}


	public int getQuid() {return quid;}

	public void setQuid(int quid) {this.quid = quid;}

	public String getTitle() {return title;}

	public void setTitle(String title) {this.title = title;}

	public int getQuizno() {return quizno;}

	public void setQuizno(int quizno) {this.quizno = quizno;}

	public String getSubject() {return subject;}

	public void setSubject(String subject) {this.subject = subject;}

	public Question getQid() {return qid;}

	public void setQid(Question qid) {this.qid = qid;}


	@Override
	public String toString() {
		return "Quiz [quid=" + quid + ", title=" + title + ", quizno=" + quizno + ", subject=" + 
subject + ", qid="
				+ qid + "]";
	}


}
```

Result

```
package com.entity;
public class Result implements Comparable<Result>{
	//private int resid;
	private String email;
	private Integer marks;

	public Result(){}

	public Result(String email2, int mark)
{super();this.email=email2;this.marks=mark;}
	// public int getResid() {return resid;}

	// public void setResid(int resid) {this.resid = resid; }
	public String getEmail() {return email;}

	public void setEmail(String email) {this.email = email;}

	public Integer getMarks() {return marks;}

	public void setMarks(Integer marks) {this.marks = marks;}

	@Override
	public String toString() {
	return "Result [email=" + email + ", marks=" + marks + "]";}

	@Override
	public int compareTo(Result r) {
		int comparemarks= r.getMarks();
		return comparemarks-this.marks;
	}
	}
```

Statistic

```
package com.entity;

import java.util.List;
import org.springframework.stereotype.Component;

@Component
public class Statistics {

	private int users;
	private List<Object> quiz;
	private int questions;

	public Statistics() {}

	public Statistics(int users, List<Object> quiz, int questions) {super();this.users = users;this.quiz =
quiz;this.questions = questions;}

	@Override
```

```
        public String toString() {
        return "Statistics [users=" + users + ", quiz=" + quiz + ", questions=" + questions + "]";}

        public int getUsers() {return users;}

        public void setUsers(int users) { this.users = users;}

        public List<Object> getQuiz() { return quiz;}

        public void setQuiz(List<Object> quiz) { this.quiz = quiz;}

        public int getQuestions() {return questions;}

        public void setQuestions(int questions) { this.questions = questions;}
}
```

Test

```
package com.entity;

import org.springframework.stereotype.Component;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;

@Component
@Entity
@Table(name="test")
public class Test {
                @Id
                private int tid;
                @ManyToOne
                @JoinColumn(referencedColumnName = "uid")
                private User userid;
                @ManyToOne
                @JoinColumn(referencedColumnName = "quid")
                private Quiz quizid;
                @ManyToOne
                @JoinColumn(referencedColumnName = "qid")
                private Question questionid;

                private int testans;
```

```
			public Test() {}

			public Test(int tid, User userid, Quiz quizid, Question questionid, int testans) {
					super();this.tid = tid;this.userid = userid;this.quizid = quizid;this.questionid =
questionid;this.testans = testans;}

			public int getTid() {return tid;}

			public void setTid(int tid) {this.tid = tid;}

			public User getUserid() {return userid;}

			public void setUserid(User userid) { this.userid = userid;}

			public Quiz getQuizid() {return quizid;}

			public void setQuizid(Quiz quizid) { this.quizid = quizid;}

			public Question getQuestionid() { return questionid;}

			public void setQuestionid(Question questionid) { this.questionid = questionid;}

			public int getTestans() {return testans;}

			public void setTestans(int testans) { this.testans = testans;}
			@Override
			public String toString() {
			return "Test [tid=" + tid + ", userid=" + userid + ", quid=" + quizid + ", questionid=" +
questionid
			+ ", testans=" + testans + "]";
			}
}
```

REPOSITORY CLASSES

Admin

```
package com.repo;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
import com.entity.Admin;
@Repository
public interface AdminRepo extends JpaRepository<Admin, Integer> {
}
```

User

```
package com.repo;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.entity.User;

@Repository
public interface UserRepo extends JpaRepository<User, Integer>{
        public User findByEmailid(String emailid);
}
```

Question

```
package com.repo;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.entity.Question;

@Repository
public interface QuestionRepo extends JpaRepository<Question, Integer> {
}
```

Quiz

```
package com.repo;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import com.entity.Quiz;
```

```
@Repository
public interface QuizRepo extends JpaRepository<Quiz, Integer>{
	@Query("select q.title,count(distinct q.quizno) from Quiz as q group by q.quizno")
	public List<Object> listOfQuiz();
}
```

Test

```
package com.repo;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import com.entity.Test;
@Repository
public interface TestRepo extends JpaRepository<Test, Integer>{

	@Query("Select t from Test as t group by t.userid")
	List<Test> getIndividual();
}
```

SERVICE CLASSES

AdminService

```
package com.service;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.entity.*;
import com.repo.AdminRepo;
import com.repo.QuestionRepo;
import com.repo.QuizRepo;
import com.repo.UserRepo;
```

```
@Service
public class AdminService {
        @Autowired
        QuestionRepo qr;
        @Autowired
        QuizRepo qur;
        @Autowired
        UserRepo ur;
        @Autowired
        Statistics stat;
        @Autowired
        AdminRepo adr;

                        public String adminLogin(Admin u){
                        Admin ad= adr.findById(1).get();
                        if(u.getUname().equals(ad.getUname())&&u.getPwd().equals(ad.getPwd()))
                        { return "Welcome admin";}
                        else{ return "invalid Credentials"; }
                        }


                        public String adminupdate(Admin a){
                        Admin ad= adr.findById(1).get();
                        ad.setUname(a.getUname());
                        ad.setPwd(a.getUname());
                        adr.saveAndFlush(ad);
                        return "Updated";}

                        public String addQuestion(Question q){
                        if(q!=null) { qr.save(q); return "question added"; }
                        else{    return "failed to add"; }
                        }
                        public String addQuiz(Quiz q){

                        if(q!=null){
                                qur.save(q);
                                return "quiz added";}
                        else{
                                return "failed to add";
                        }
                        }
                        public List<Quiz> viewAllQuiz(){
                        return qur.findAll();
```

```
                              }
                              public Statistics quizInfo(){
                              stat.setUsers(ur.findAll().size());
                              stat.setQuestions(qr.findAll().size());
                              stat.setQuiz(qur.listOfQuiz());
                              return stat;
                              }
}
```

UserService

```
package com.service;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.entity.*;
import com.repo.QuizRepo;
import com.repo.TestRepo;
import com.repo.UserRepo;



@Service
public class UserService {


            List<Result> finalList=new ArrayList<>();

            @Autowired
            UserRepo ur;
            @Autowired
            QuizRepo qr;
            @Autowired
            TestRepo tr;
            @Autowired
            User u;
            @Autowired
            Test t;

            Result r= new Result();
```

```
public String userLogin(String email,String password){
u=ur.findByEmailid(email);
if(u!=null){
        if(u.getEmailid().equals(email)&&u.getPassword().equals(password))
                { return "login sucessfull";}
        else return "invalid credentials";              }
else{return "User not found"; }
}

public String userRegister(User u){
if(ur.findByEmailid(u.getEmailid())==null){
        ur.save(u);
        return "registered";}
else{return "User already exists"; }
}

public List<Object> viewAllQuiz(){
return qr.listOfQuiz();
}

public String takeTest(Test t){
if(t!=null){
        tr.save(t);
        return "submitted";}
else{   return "submission failed"; }
}

public List<Test> getTestList(){
return tr.findAll();
}

public List<Result> result(){
String email=""; int mark=0;
List<Test> obj=tr.findAll();
List<User> u= ur.findAll();
for (User user : u) {
        mark=0;
        email=user.getEmailid();
        System.out.println(user.getEmailid());

        for(Test ob :obj){
        if(user.getUid()==ob.getUserid().getUid()){
                if(ob.getTestans()==ob.getQuestionid().getAns()){
```

```
                                                  mark++;}
                                        System.out.println("inside"+mark);
                              }
                              }
                              System.out.println("outside"+mark);

                              finalList.add(new Result(email,mark));
                              }
                    System.out.println("final :"+mark);
                    Collections.sort(finalList);
                    return finalList;
                    }
}
```

CONTROLLER CLASS

```
package com.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.entity.Admin;
import com.entity.Question;
import com.entity.Quiz;
import com.entity.Result;
import com.entity.Statistics;
import com.entity.Test;
import com.entity.User;
import com.service.AdminService;
import com.service.UserService;

@RestController
@RequestMapping("/mcq")
public class MainController {
                    @Autowired
                    UserService us;
```

```
		@Autowired
		AdminService as;

		//http://localhost:8080/mcq/userLogin
		@GetMapping(value="/userLogin", consumes =
MediaType.APPLICATION_JSON_VALUE)
		public String userLogin(@RequestBody User u){
		return us.userLogin(u.getEmailid(), u.getPassword());
		}

		//http://localhost:8080/mcq/userRegister
		@PostMapping(value="/userRegister", consumes =
MediaType.APPLICATION_JSON_VALUE)
		public String userRegiter(@RequestBody User u){
		return us.userRegister(u);
		}

		//http://localhost:8080/mcq/adminLogin
		@PostMapping(value="/adminLogin", consumes =
MediaType.APPLICATION_JSON_VALUE)
		public String adminLogin(@RequestBody Admin adm){
		return as.adminLogin(adm);
		}

		//http://localhost:8080/mcq/adminupdate
		@PostMapping(value="adminupdate", consumes =
MediaType.APPLICATION_JSON_VALUE)
		public String adminUpdate(@RequestBody Admin adm){
		return as.adminupdate(adm);
		}


		//http://localhost:8080/mcq/addQuestions
		@PostMapping(value="/addQuestions" , consumes =
MediaType.APPLICATION_JSON_VALUE)
		public String addQuestion(@RequestBody Question q){
		return as.addQuestion(q);
		}

		//http://localhost:8080/mcq/addQuiz
		@PostMapping(value="/addQuiz"  , consumes =
MediaType.APPLICATION_JSON_VALUE)
		public String addQuiz(@RequestBody Quiz qz){
```

```
                    return as.addQuiz(qz);
                    }

                    //http://localhost:8080/mcq/viewAllQuiz
                    @GetMapping(value="/viewAllQuiz" , produces=
MediaType.APPLICATION_JSON_VALUE)
                    public List<Quiz> viewAllQuiz(){
                    return as.viewAllQuiz();
                    }

                    //http://localhost:8080/mcq/quizinfo
                    @GetMapping(value="/quizinfo" , produces=
MediaType.APPLICATION_JSON_VALUE)
                    public Statistics quizinfo()
                    {
                    return as.quizInfo();
                    }

                    //http://localhost:8080/mcq/viewQuiz
                    @GetMapping(value="/viewQuiz", produces=
MediaType.APPLICATION_JSON_VALUE)
                    public List<Object> viewQuiz()
                    {
                    return us.viewAllQuiz();
                    }
                    //http://localhost:8080/mcq/takeTest
                    @PostMapping(value="/takeTest" , consumes =
MediaType.APPLICATION_JSON_VALUE)
                    public String takeTest(@RequestBody Test t)
                    {
                    return us.takeTest(t);
                    }
                    //http://localhost:8080/mcq/getAllTest
                    @GetMapping(value="/getAllTest" , produces=
MediaType.APPLICATION_JSON_VALUE)
                    public List<Test> getAllTest()
                    {
                    return us.getTestList();
                    }
                    //http://localhost:8080/mcq/getresult
                    @GetMapping(value="/getresult" , produces=
MediaType.APPLICATION_JSON_VALUE)
                    public List<Result> getresult()
                    {
                    return us.result();
                    }
}
```

Application.properties

```
#SQL Database
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/phasethree
spring.datasource.username=root
spring.datasource.password={database-password}
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.show-sql: true
spring.jpa.properties.hibernate.format_sql=true
logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.type=TRACE

#JSP view resolver support
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
```

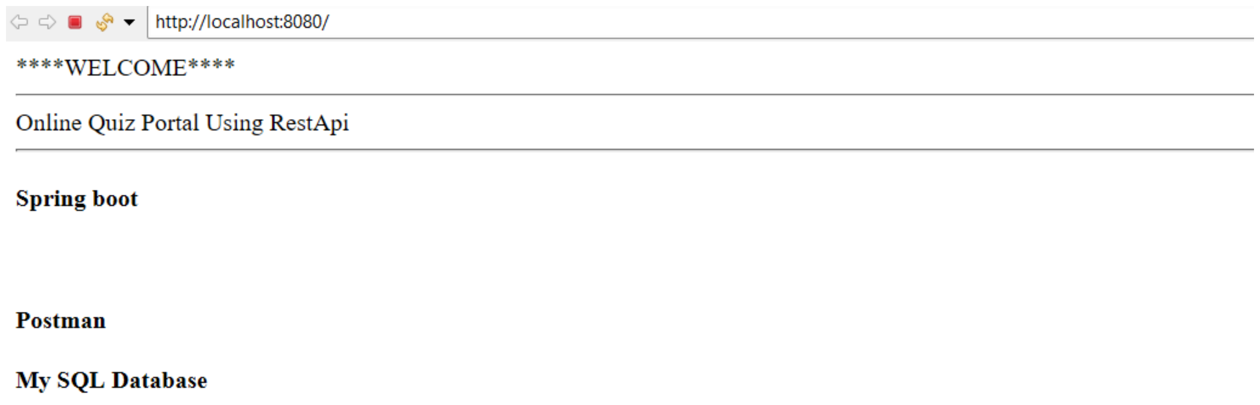
SQL- queries

```
create database phasethree
use phasethree

-- admin table
create table admin(ID int primary key, Uname varchar(255), Pwd varchar(80))
select * from admin

--user table
create table Userss(uid int primary key,Email varchar(255) ,upwd varchar(80),Ph_NO BigInt )
select * from Userss

--questions table
create table question(qid int primary key,question varchar(255),opt1 varchar(150),opt2 varchar(150) ,opt3
varchar(150) ,opt4 varchar(150),ans int)
select * from question
```

--quiz table
create table quiz(quid int primary key ,quizno int ,Subject varchar(80) ,title varchar(50),qid int ,foreign key(qid) references question(qid))
select * from quiz

--test table
create table test(tid int primary key, userid int ,quizid int ,questionid int ,testans int , foreign key(userid) references Userss(uid),foreign key(quizid) references quiz(quid),foreign key(questionid) references question(qid))
select * from test

OUTPUTS


http://localhost:8080/

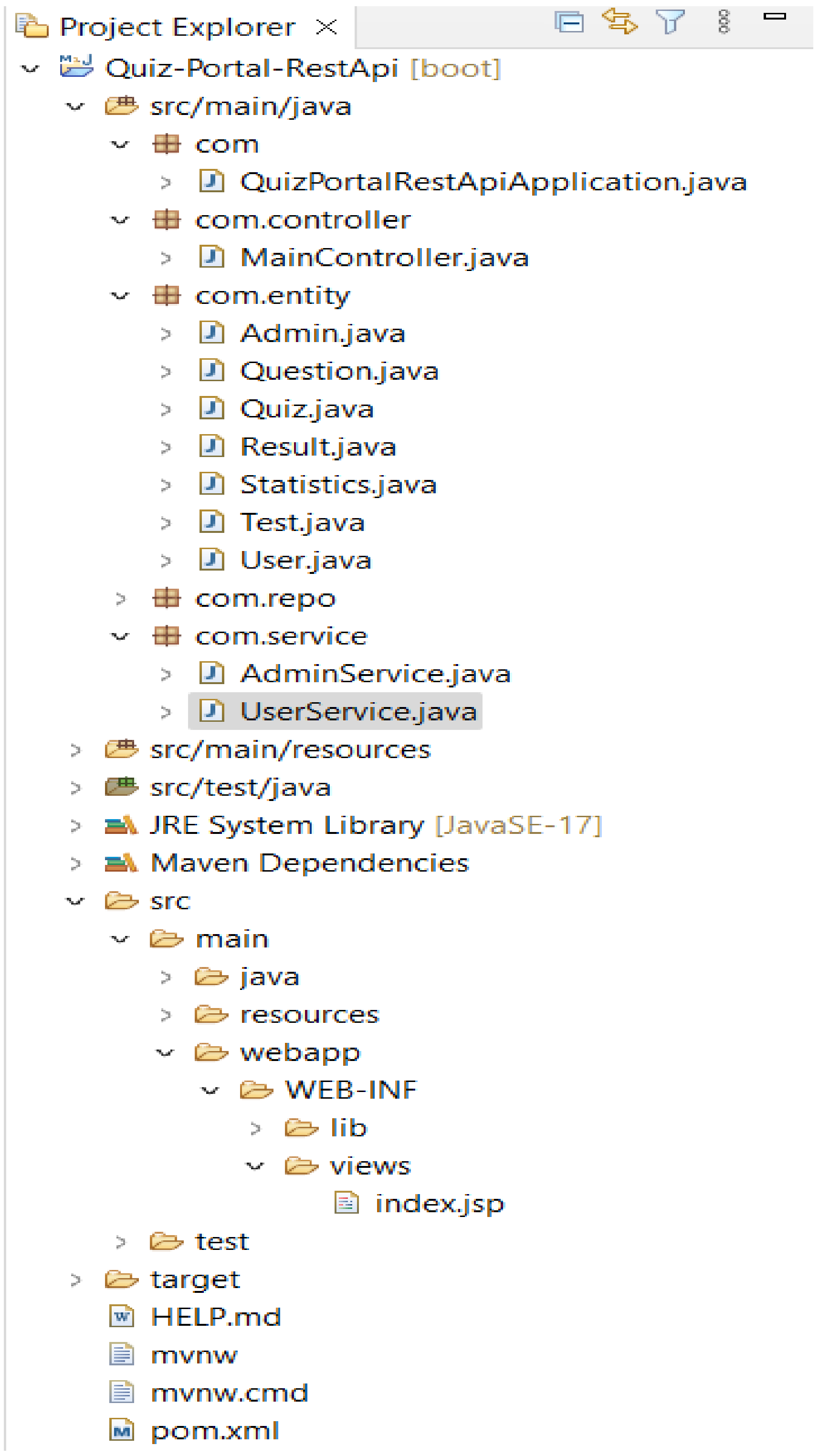
****WELCOME****

Online Quiz Portal Using RestApi

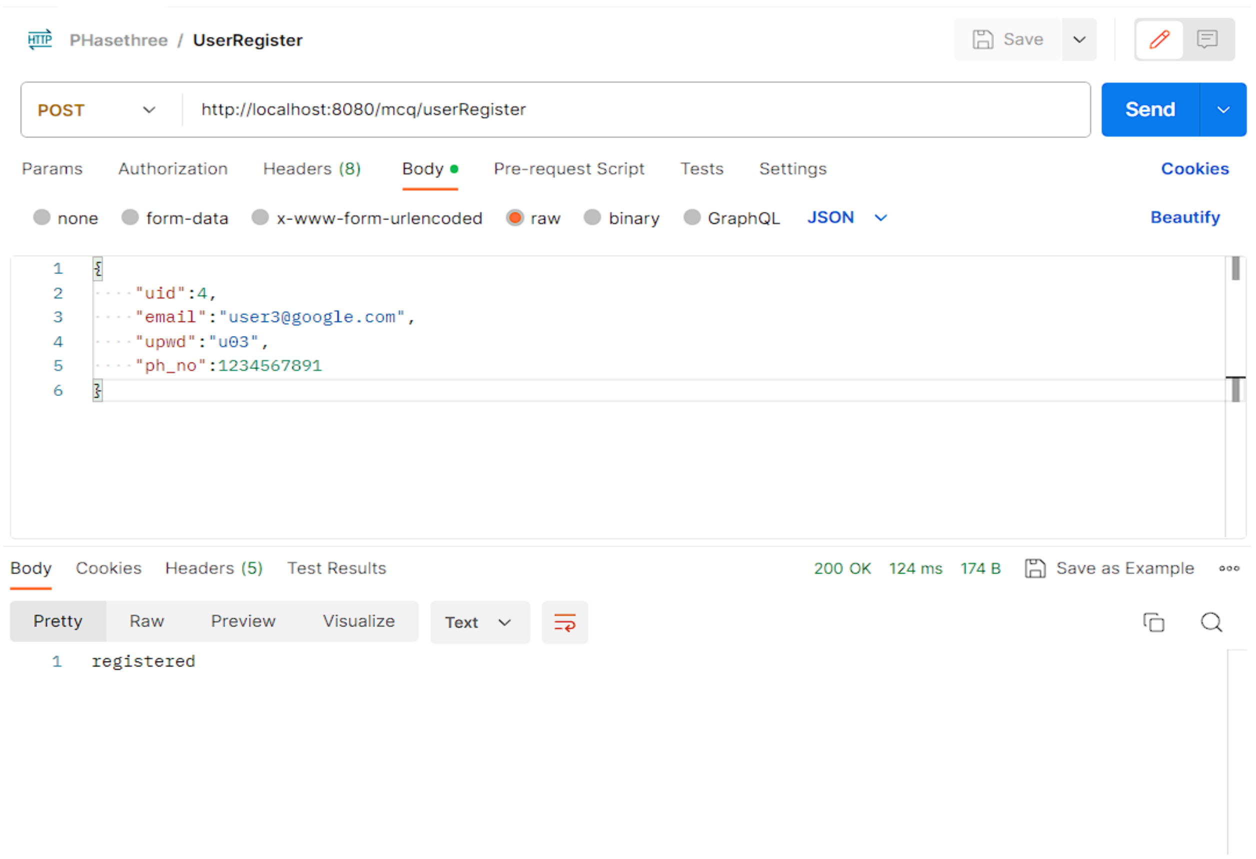
Spring boot

Postman

My SQL Database



```
Project Explorer
v Quiz-Portal-RestApi [boot]
  v src/main/java
    v com
      > QuizPortalRestApiApplication.java
    v com.controller
      > MainController.java
    v com.entity
      > Admin.java
      > Question.java
      > Quiz.java
      > Result.java
      > Statistics.java
      > Test.java
      > User.java
    > com.repo
    v com.service
      > AdminService.java
      > UserService.java
  > src/main/resources
  > src/test/java
  > JRE System Library [JavaSE-17]
  > Maven Dependencies
  v src
    v main
      > java
      > resources
      v webapp
        v WEB-INF
          > lib
          v views
              index.jsp
    > test
  > target
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
```


PHasethree / UserRegister
Save
Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify
Body Cookies Headers (5) Test Results
Save as Example
Pretty Raw Preview Visualize Text

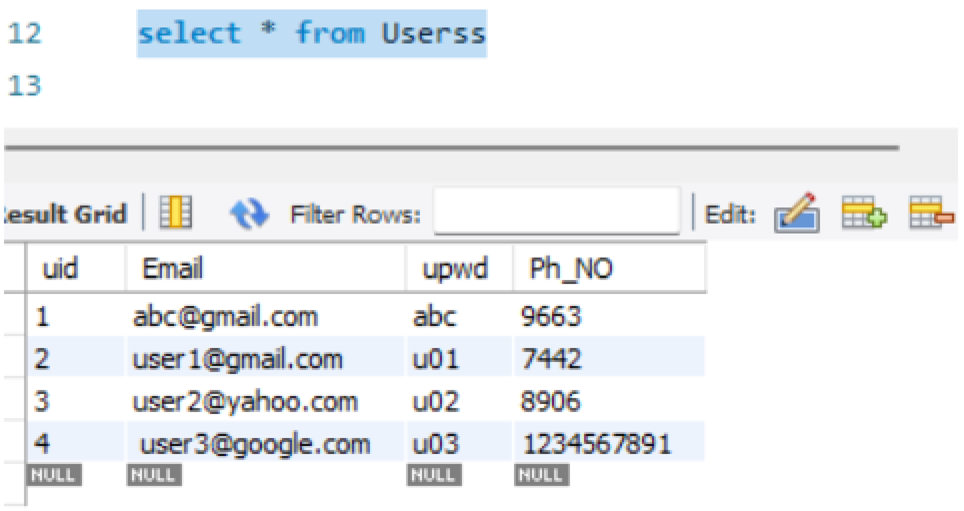
POST http://localhost:8080/mcq/userRegister

Send

```
{
    "uid":4,
    "email":"user3@google.com",
    "upwd":"u03",
    "ph_no":1234567891
}
```

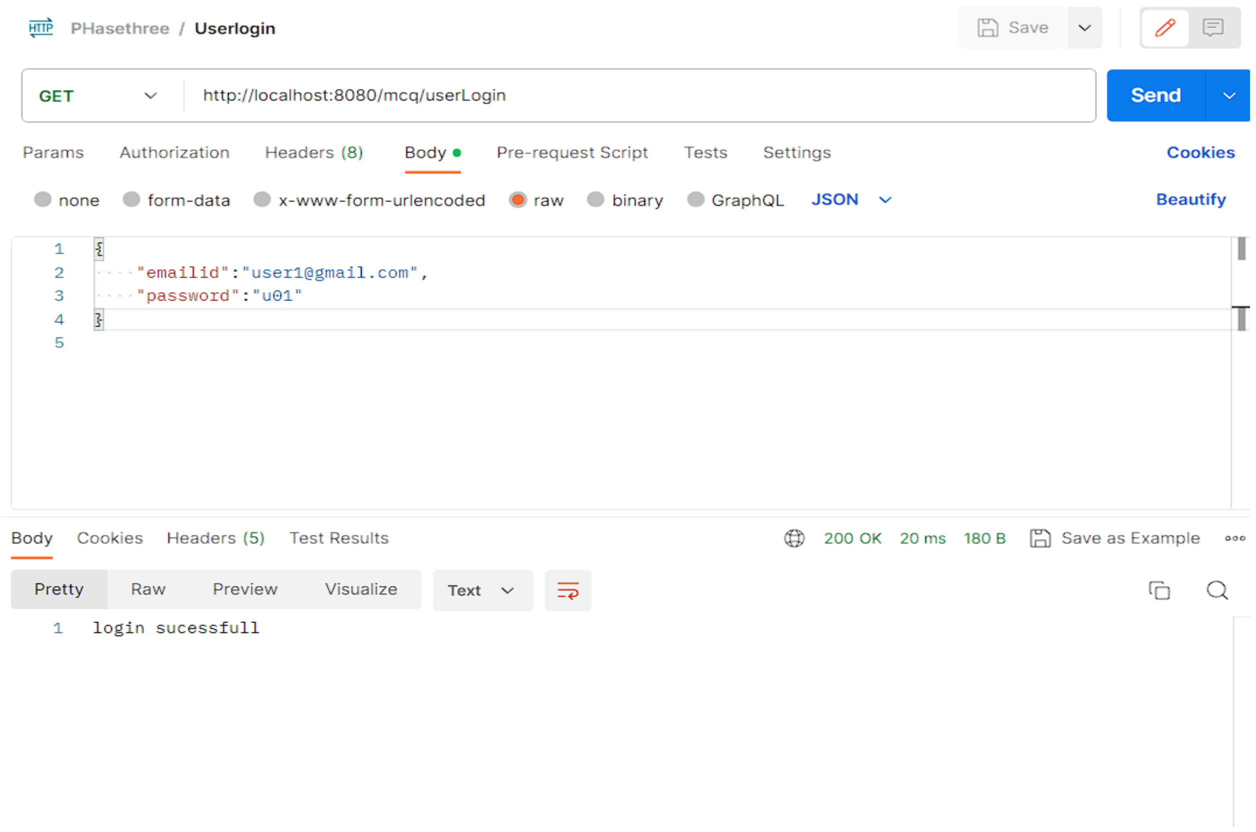
200 OK 124 ms 174 B

```
registered
```



```
select * from Userss
```

uid	Email	upwd	Ph_NO
1	abc@gmail.com	abc	9663
2	user1@gmail.com	u01	7442
3	user2@yahoo.com	u02	8906
4	user3@google.com	u03	1234567891
NULL	NULL	NULL	NULL

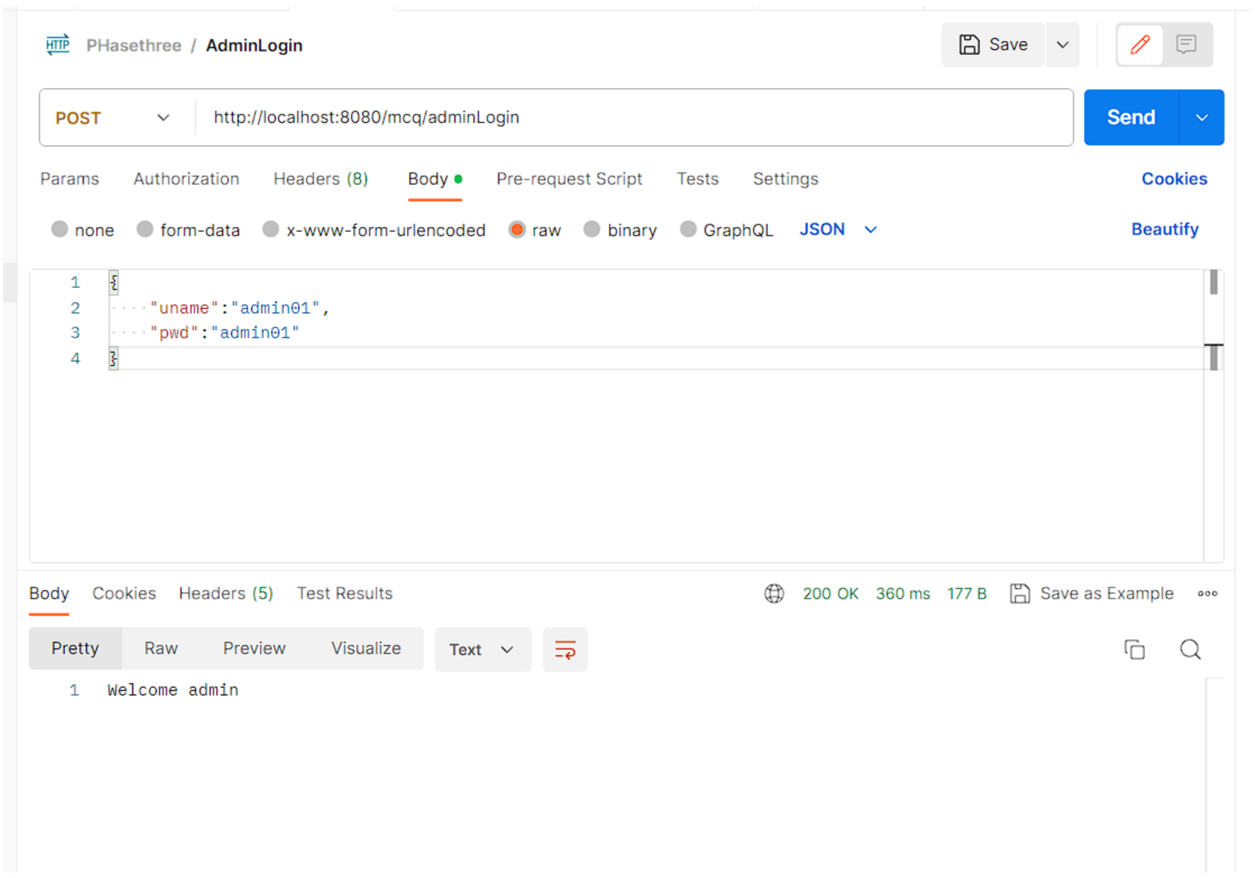


GET http://localhost:8080/mcq/userLogin

```
{
    "emailid":"user1@gmail.com",
    "password":"u01"
}
```

200 OK 20 ms 180 B

```
login sucessfull
```

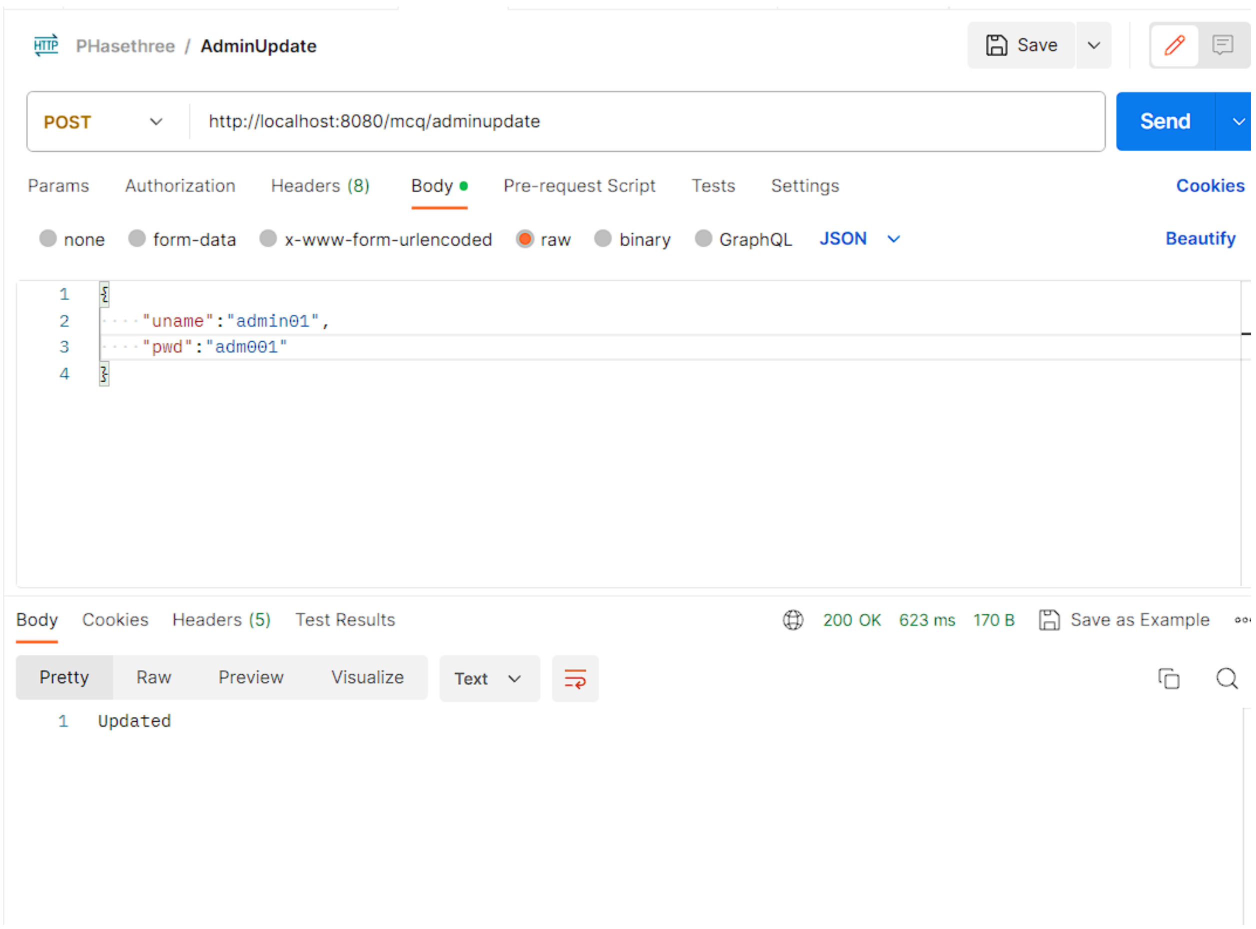


POST http://localhost:8080/mcq/adminLogin

```
{
    "uname":"admin01",
    "pwd":"admin01"
}
```

200 OK 360 ms 177 B

```
Welcome admin
```



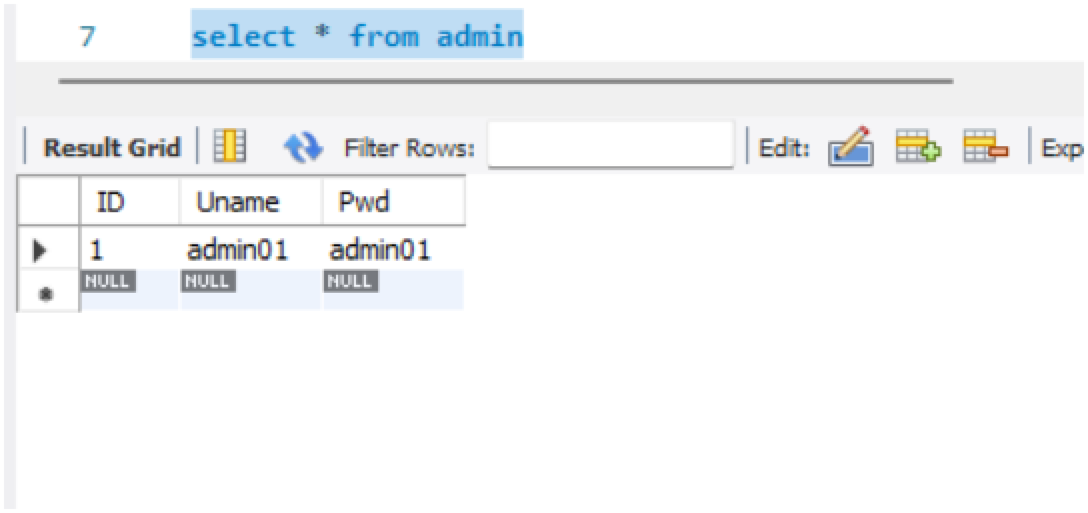
PHasethree / AdminUpdate

POST http://localhost:8080/mcq/adminupdate

```
{
    "uname":"admin01",
    "pwd":"adm001"
}
```

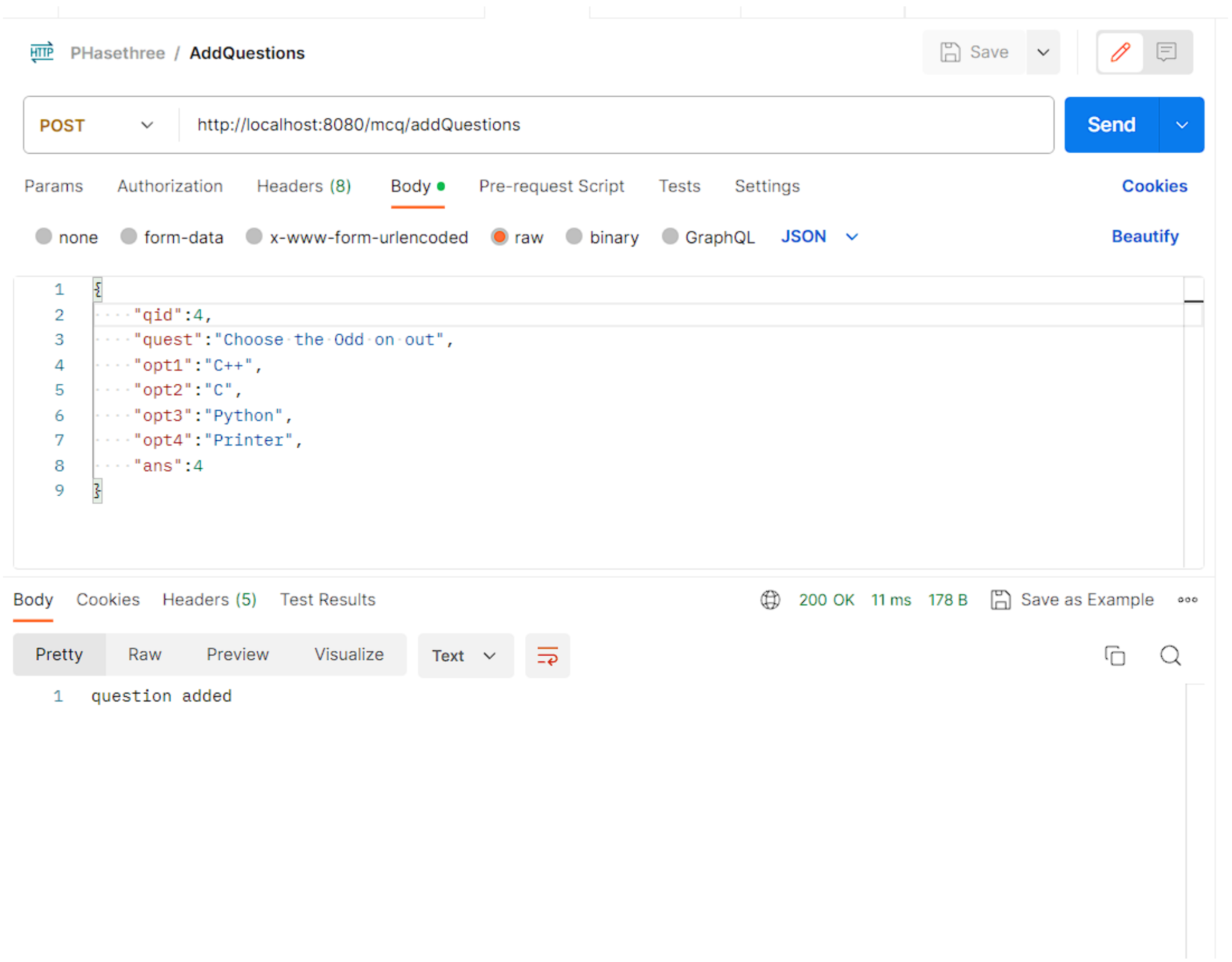
200 OK 623 ms 170 B

```
Updated
```



```
select * from admin
```

ID	Uname	Pwd
1	admin01	admin01
NULL	NULL	NULL


PHasethree / AddQuestions
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies
Beautify
Body Cookies Headers (5) Test Results
Save as Example
Pretty Raw Preview Visualize

POST http://localhost:8080/mcq/addQuestions

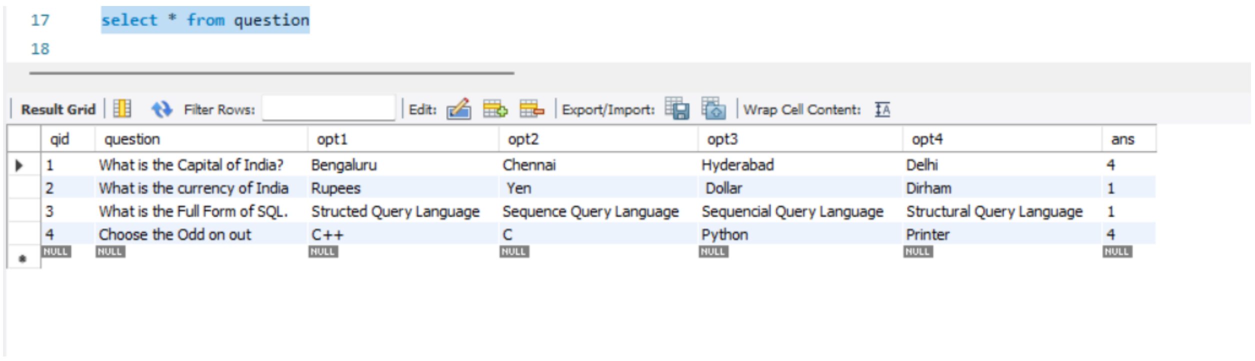
none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON

```
{
    "qid":4,
    "quest":"Choose the Odd on out",
    "opt1":"C++",
    "opt2":"C",
    "opt3":"Python",
    "opt4":"Printer",
    "ans":4
}
```

200 OK 11 ms 178 B

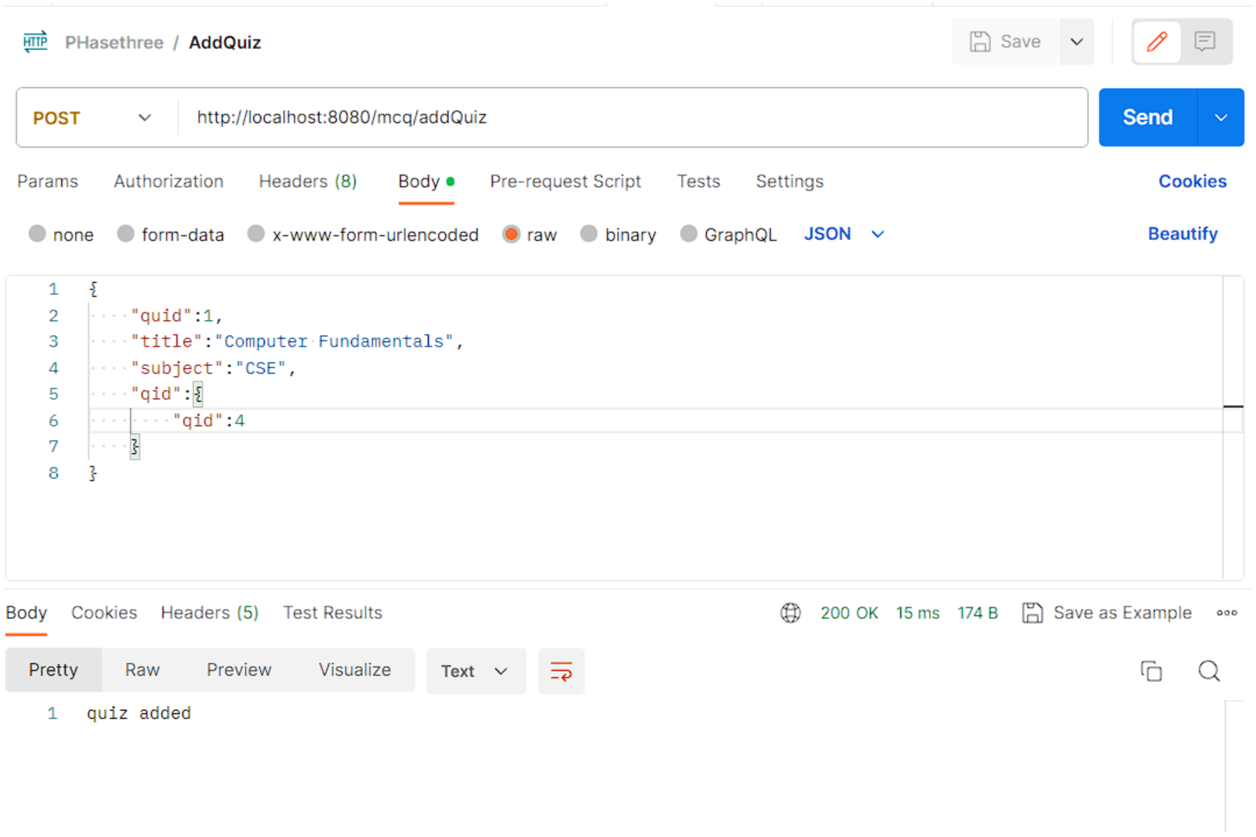
Text

```
question added
```



```
select * from question
```

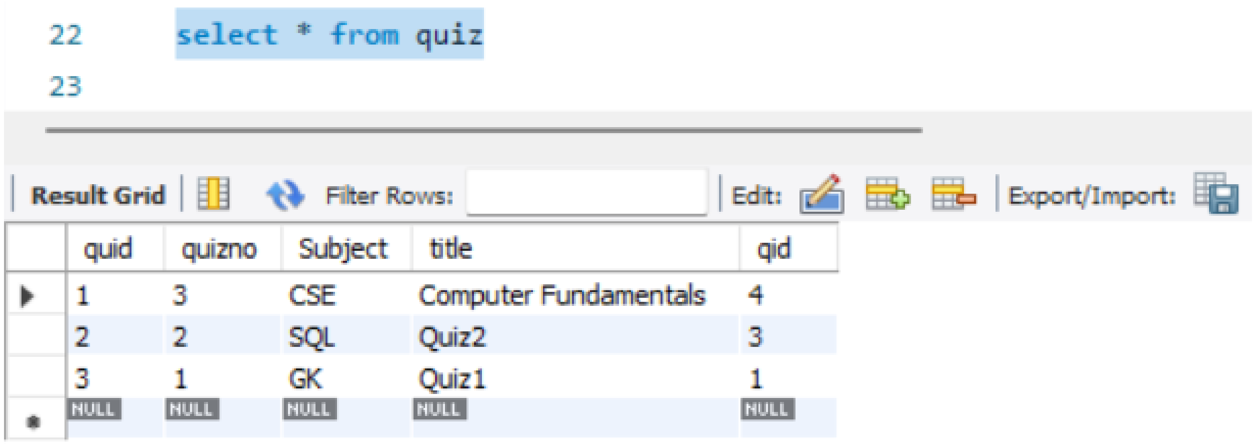
qid	question	opt1	opt2	opt3	opt4	ans
1	What is the Capital of India?	Bengaluru	Chennai	Hyderabad	Delhi	4
2	What is the currency of India	Rupees	Yen	Dollar	Dirham	1
3	What is the Full Form of SQL.	Structed Query Language	Sequence Query Language	Sequencial Query Language	Structural Query Language	1
4	Choose the Odd on out	C++	C	Python	Printer	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL



POST http://localhost:8080/mcq/addQuiz

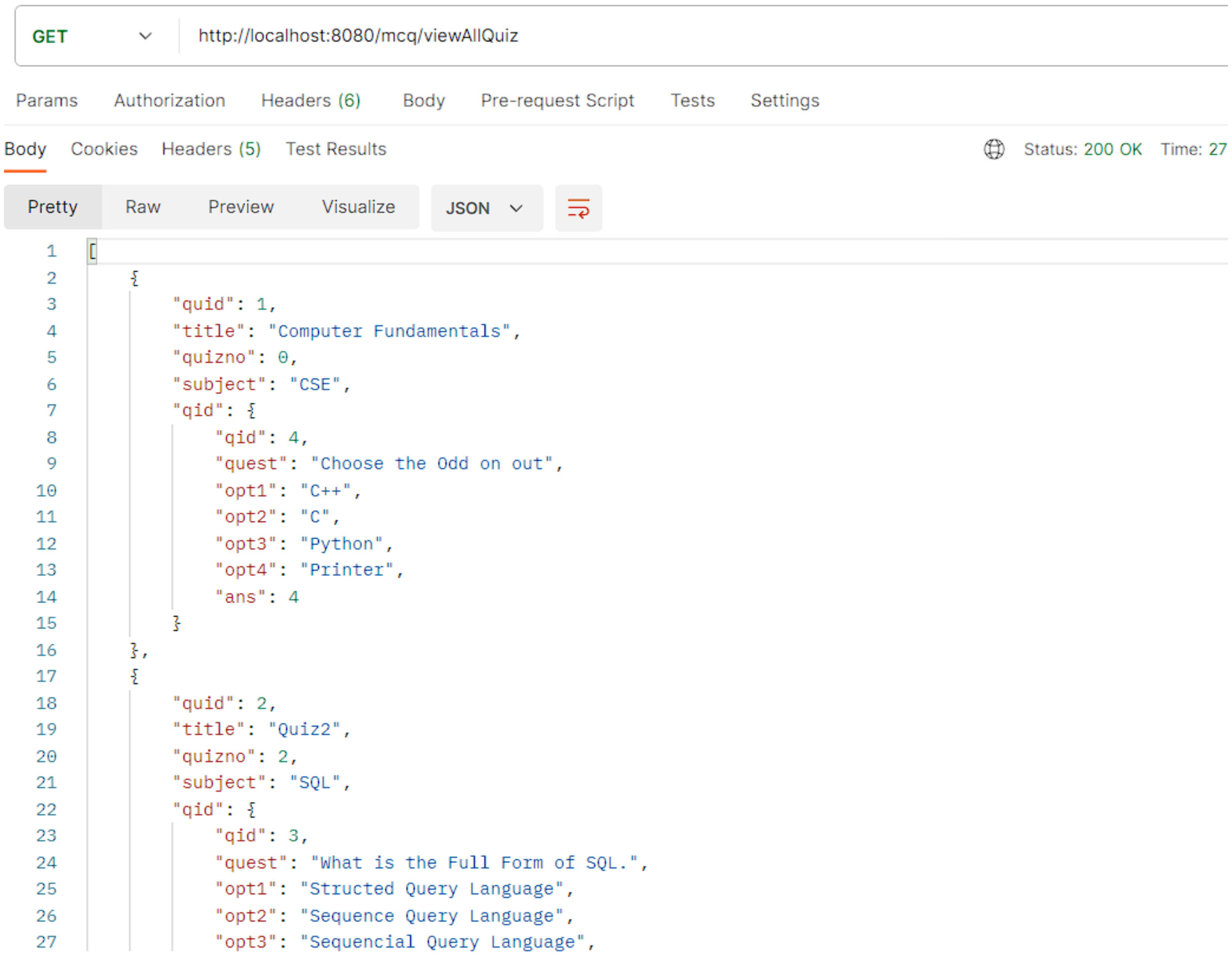
```
{
    "quid":1,
    "title":"Computer Fundamentals",
    "subject":"CSE",
    "qid":{
        "qid":4
    }
}
```

```
quiz added
```



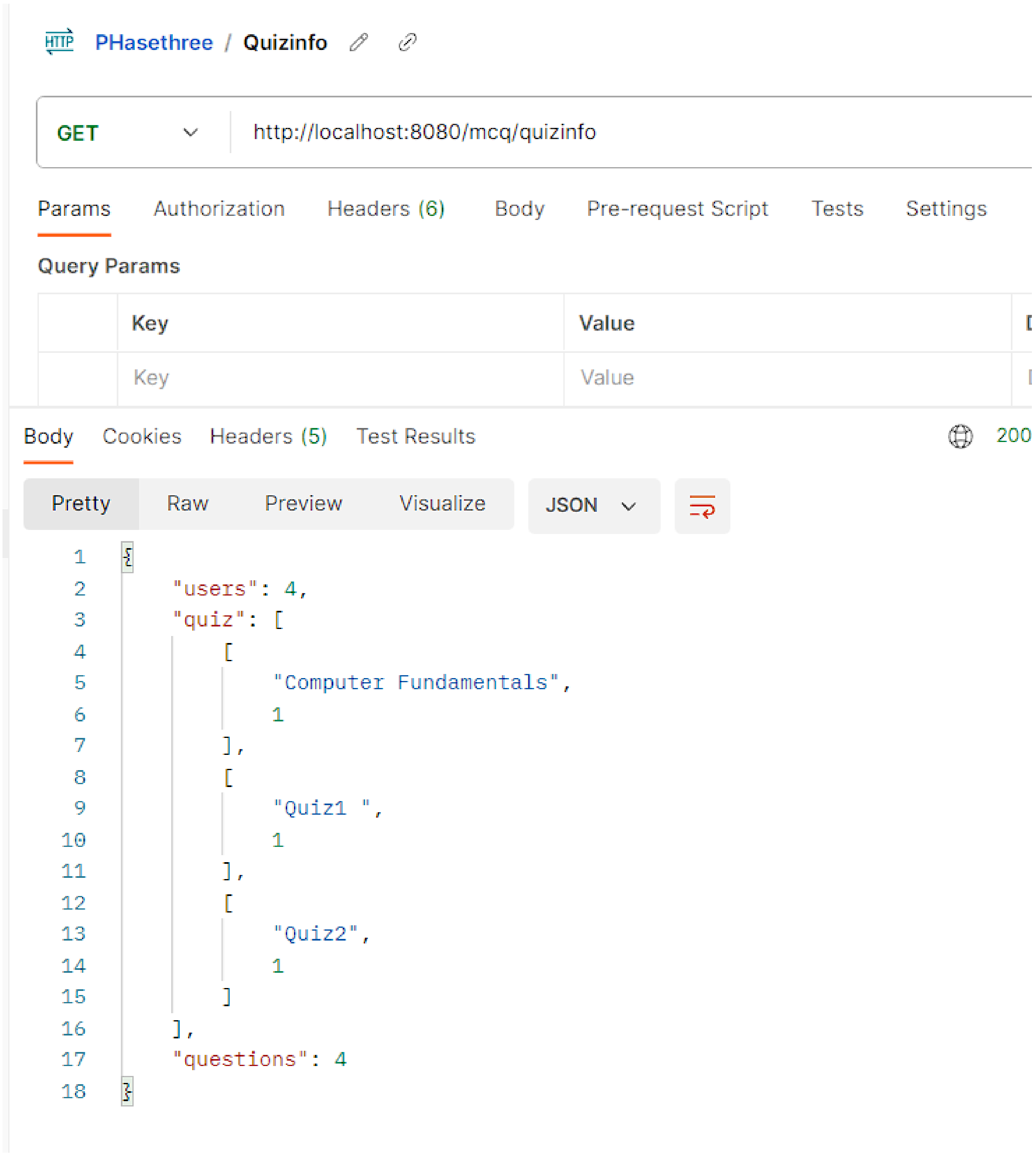
```
22      select * from quiz
23
```

quid	quizno	Subject	title	qid
1	3	CSE	Computer Fundamentals	4
2	2	SQL	Quiz2	3
3	1	GK	Quiz1	1
NULL	NULL	NULL	NULL	NULL



GET http://localhost:8080/mcq/viewAllQuiz

```
[
    {
        "quid": 1,
        "title": "Computer Fundamentals",
        "quizno": 0,
        "subject": "CSE",
        "qid": {
            "qid": 4,
            "quest": "Choose the Odd on out",
            "opt1": "C++",
            "opt2": "C",
            "opt3": "Python",
            "opt4": "Printer",
            "ans": 4
        }
    },
    {
        "quid": 2,
        "title": "Quiz2",
        "quizno": 2,
        "subject": "SQL",
        "qid": {
            "qid": 3,
            "quest": "What is the Full Form of SQL.",
            "opt1": "Structed Query Language",
            "opt2": "Sequence Query Language",
            "opt3": "Sequencial Query Language",
        "opt4": "Structural Query Language",
        "ans": 1
    }
},
{
    "quid": 3,
    "title": "Quiz1 ",
    "quizno": 1,
    "subject": "GK",
    "qid": {
        "qid": 1,
        "quest": "What is the Capital of India?",
        "opt1": "Bengaluru",
        "opt2": "Chennai",
        "opt3": "Hyderabad",
        "opt4": "Delhi",
        "ans": 4
    }
}
```

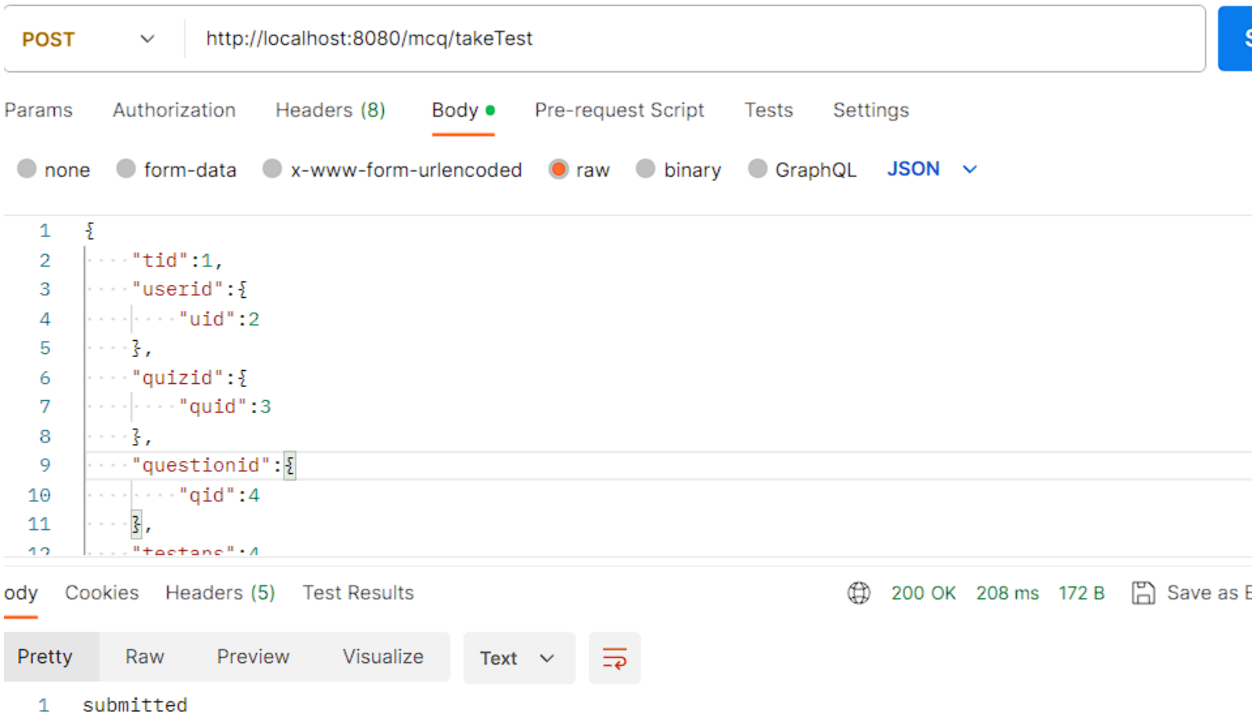


http://localhost:8080/mcq/quizinfo

Query Params

	Key	Value
	Key	Value

```
{
    "users": 4,
    "quiz": [
        [
            "Computer Fundamentals",
            1
        ],
        [
            "Quiz1 ",
            1
        ],
        [
            "Quiz2",
            1
        ]
    ],
    "questions": 4
}
```

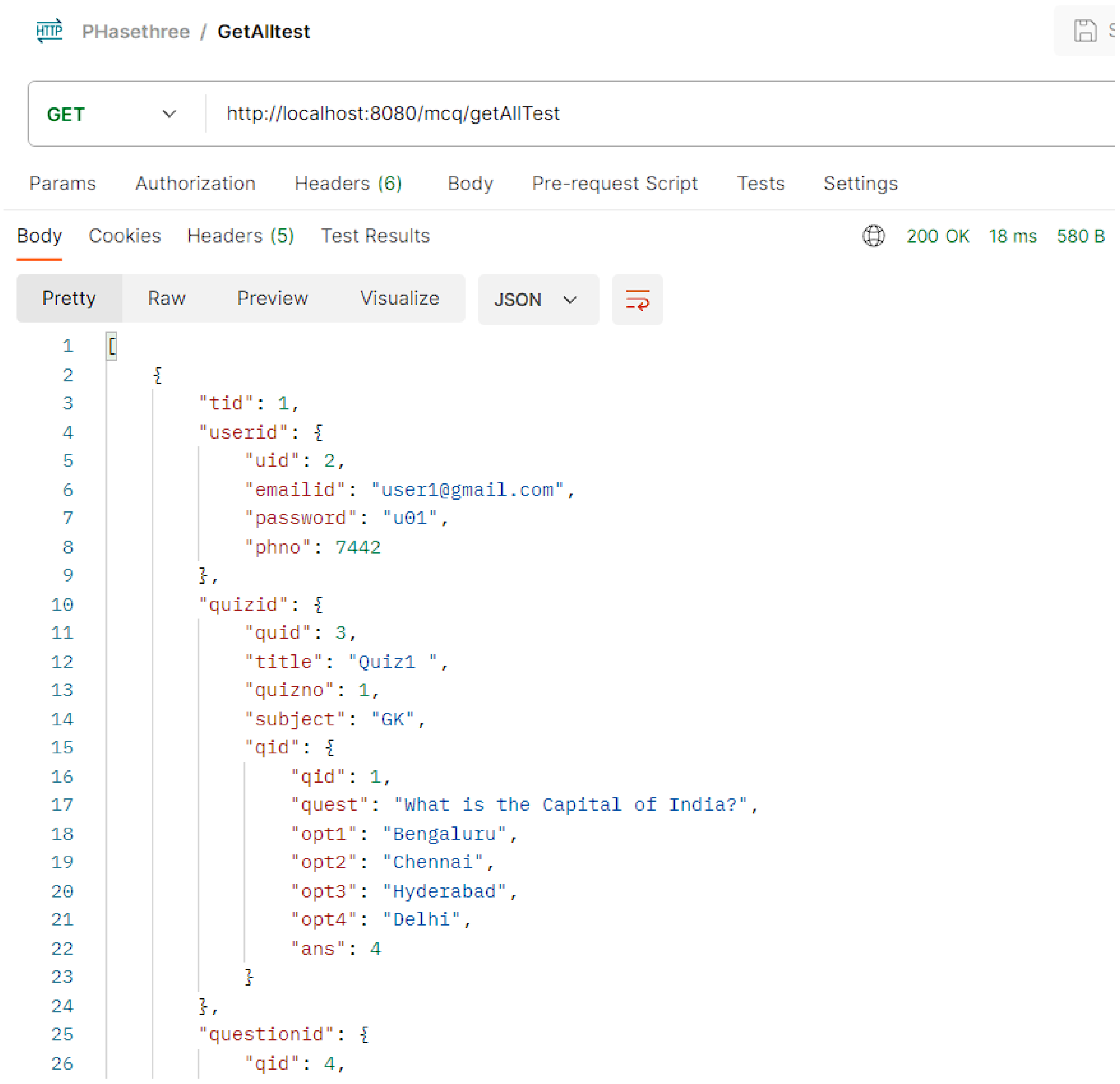


http://localhost:8080/mcq/takeTest

```
{
    "tid":1,
    "userid":{
        "uid":2
    },
    "quizid":{
        "quid":3
    },
    "questionid":{
        "qid":4
    },
    "testans":4
```

200 OK 208 ms 172 B

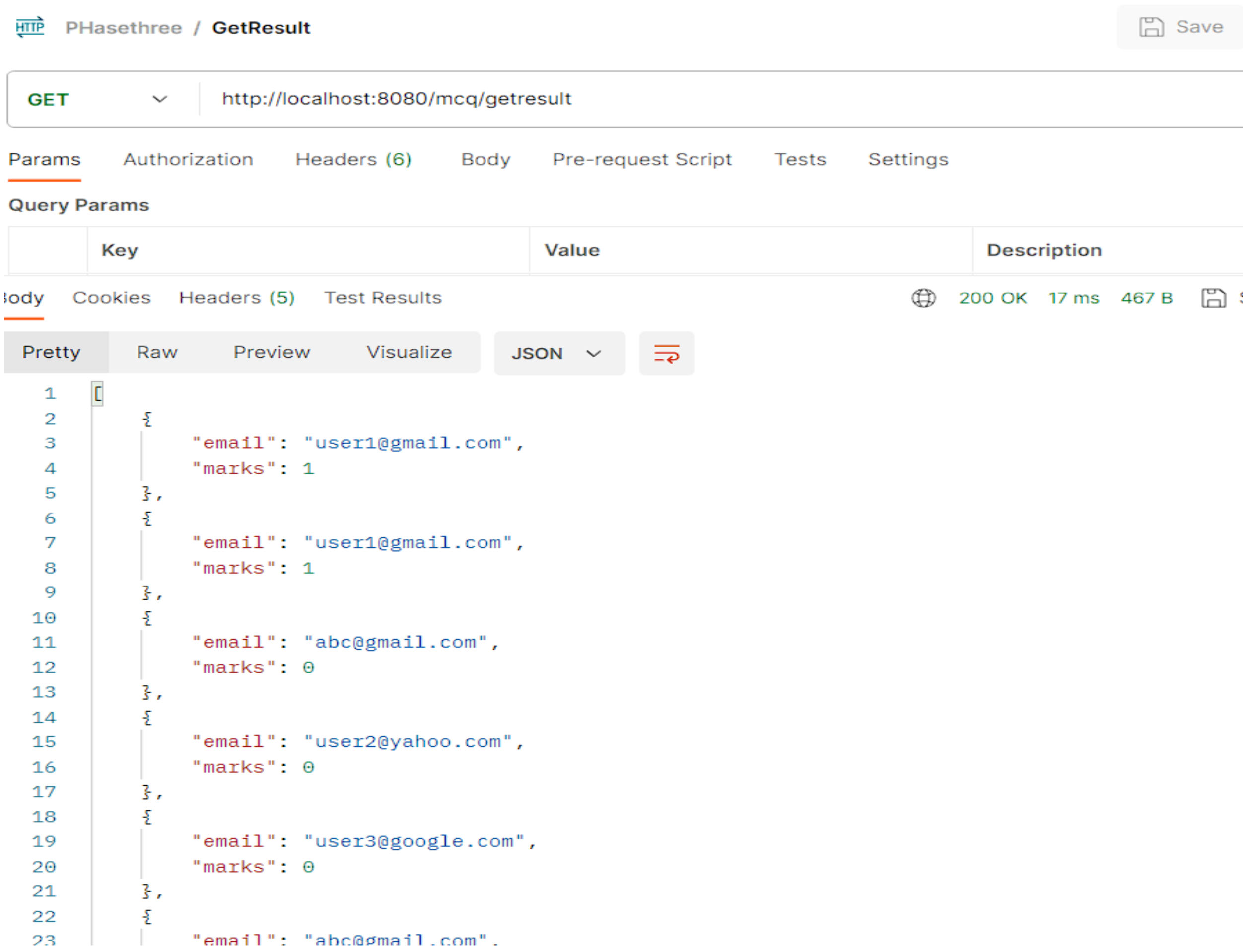
```
submitted
```



GET http://localhost:8080/mcq/getAllTest

200 OK 18 ms 580 B

```
[
    {
        "tid": 1,
        "userid": {
            "uid": 2,
            "emailid": "user1@gmail.com",
            "password": "u01",
            "phno": 7442
        },
        "quizid": {
            "quid": 3,
            "title": "Quiz1 ",
            "quizno": 1,
            "subject": "GK",
            "qid": {
                "qid": 1,
                "quest": "What is the Capital of India?",
                "opt1": "Bengaluru",
                "opt2": "Chennai",
                "opt3": "Hyderabad",
                "opt4": "Delhi",
                "ans": 4
            }
        },
        "questionid": {
            "qid": 4,
```



GET http://localhost:8080/mcq/getresult

200 OK 17 ms 467 B

```
[
    {
        "email": "user1@gmail.com",
        "marks": 1
    },
    {
        "email": "user1@gmail.com",
        "marks": 1
    },
    {
        "email": "abc@gmail.com",
        "marks": 0
    },
    {
        "email": "user2@yahoo.com",
        "marks": 0
    },
    {
        "email": "user3@google.com",
        "marks": 0
    },
    {
        "email": "abc@gmail.com",
```